Requirements

Description

- Issue: Requirements Creep

- Golden rule: Bring the money
- Shoot the user (freeze the baseline)

- New acquisition vs. legacy systems
- Short term/small scope vs. long term/large scope requirements

Short Term Actions/Who?

- Build for recognition of requirements creep
  - Modular systems
  - Block buys
  - Design margin
- Integrate user and Program Office

Good/Best Practices

- Use SE Discipline at all requirements decision levels (i.e. a Chief Systems Engineer at each level)
- User groups
- Stakeholder-Program Office relationship

Barriers

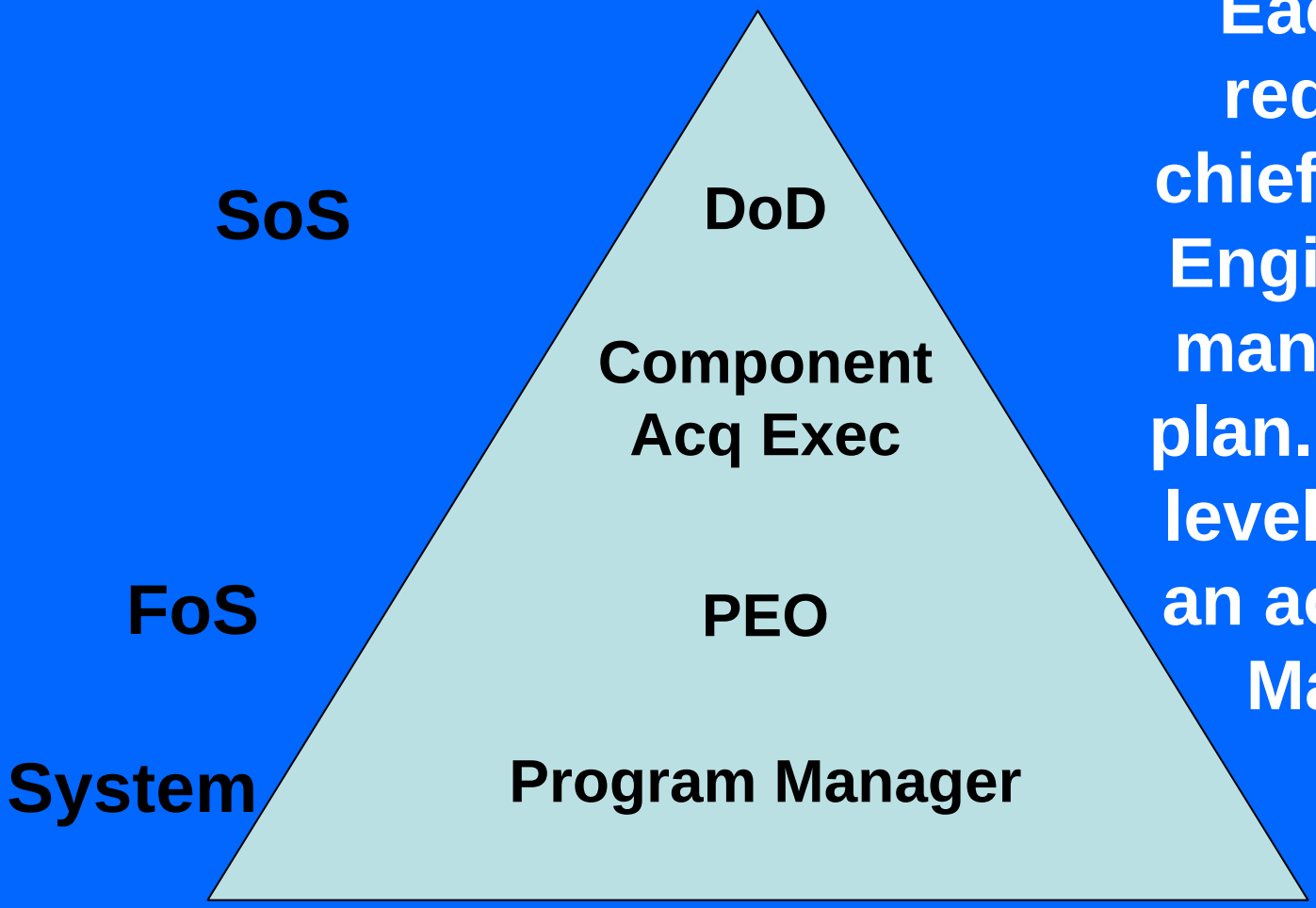
- “Bank robbers” to the budget
- Budget cycle and requirements cycle out of synch

Long Term Actions/Who?

- Joint spec – cross walked with users buy-in
- Risk Acceptance/Responsibility (AF OSS&E)
- Disciplined use of review process (honesty)
- Responsive to customer
- Educate user to take SE seriously
- Synch budget cycle to requirements cycle

Systems of Systems
&
Family of Systems Engineering

Impact



Each level requires a chief Systems Engineer and management plan. The SoS level requires an active SoS Manager

Solutions

- Support the JCIDS process
- Establish an SoS Manager
- Require an SoS Engineering Plan
- Budget at the SoS level

Barriers

- Title X and Title 50, USC authorities not aligned
- PM evaluation does not align with this process
- Asking PM to defer part of their authority to another organization – *runs counter to their training and normal function*
- Lack of SoS requirements

Good/Best Practices

- Ensure reviews use a disciplined process
- Be responsive and communicative with customer.
- Set of requirements IPTs
  - Users-Stakeholders
  - User-PMO-Stakeholders
  - PMO-Contractor

Works well when there's 1 user, 1 PM, and 1 Contractor

Good/Best Practices

- Processes that are needed at the SoS level exist/are known. They likely need to be rescaled or modified, but no need to reinvent the wheel here.
- Establish/report on -- synchronization is key at the SoS level
- Work harder to make metrics more effective so the user understands value of system.
- Research ways to reduce the development cycle.

Dealing with GIG / Net-Centric

Description

- Management structure
- Do we have the right process tools? Will process/tools give us what we want in the end?

Impact

- Shorten the kill chain

Barriers

- We don’t understand the boundaries
- Lack of common understanding on how GIG operates
- We’re stove-piped -- not set up to do this
- Need SE Tools to perform end-to-end simulations to accurately account for complex behaviors

Good/Best Practices

- Formal Architecture / Development

Short Term Actions/Who?

- Common focal point for the GIG (i.e. DoD system level architect)
- Rethink the downward net-centric requirements until the architecture is developed.

Long Term Actions/Who?

- Develop an organizational construct to develop this infrastructure (DoD? Service?)